

03

Aba Fornecedores

Transcrição

Agora que já criamos a aba de Agendamentos, podemos criar a de Fornecedores.

Vamos abrir o arquivo `fornecedores.html`. Ele tem a tag básica `<ion-content>` e dentro, a do componente de lista que adicionamos. Ele também criou três listas por padrão.

```
<ion-view title="Fornecedores" id="page3">
  <ion-content padding="true" class="has-header">
    <ion-list id="fornecedores-list1">
      <ion-item id="fornecedores-list-item4">Item 01</ion-item>
      <ion-item id="fornecedores-list-item5">Item 2</ion-item>
      <ion-item id="fornecedores-list-item6">Item 3</ion-item>

    </ion-list>
  </ion-content>
</ion-view>
```

Começaremos já apagando os últimos dois itens. Nós também queremos que eles sejam dinâmicos. Para isto, vamos pegar uma lista da `controller`, que será percorrida para usar vários itens.

No arquivo `controller.js`, teremos que injetar novamente o `$scope`, apesar de já termos usado no `AgendamentoController`:

```
.controlle('FornecedorController', function($scope){
  $scope.fornecedores = [];
})
```

Precisaremos, agora, do `Service`. No arquivo `services.js`, adicionaremos `obterFornecedores`:

```
obterFornecedores : function(){
  return $http.get(url + "/fornecedores").then(function(response){
    return response.data;
  })
}
```

Retornaremos uma chamada HTTP, não precisaremos injetar o `$http` novamente no `.service`, porque ainda estamos novamente. E concatenamos a `url` com o `/fornecedores`. No fim, retornamos o `response.data`

Agora no `controller.js`, poderemos injetar o `GerenciadorService` e usá-lo mais abaixo:

```
.controlle('FornecedorController', function($scope){
  $scope.fornecedores = [];
  GerenciadorService.obterFornecedores().then(function(dados){
    $scope.fornecedores = dados;
  })
})
```

Adicionamos uma `function()` com callback. Utilizamos a variável `fornecedores` e recebemos os dados. Agora que populamos a variável, vamos para tela.

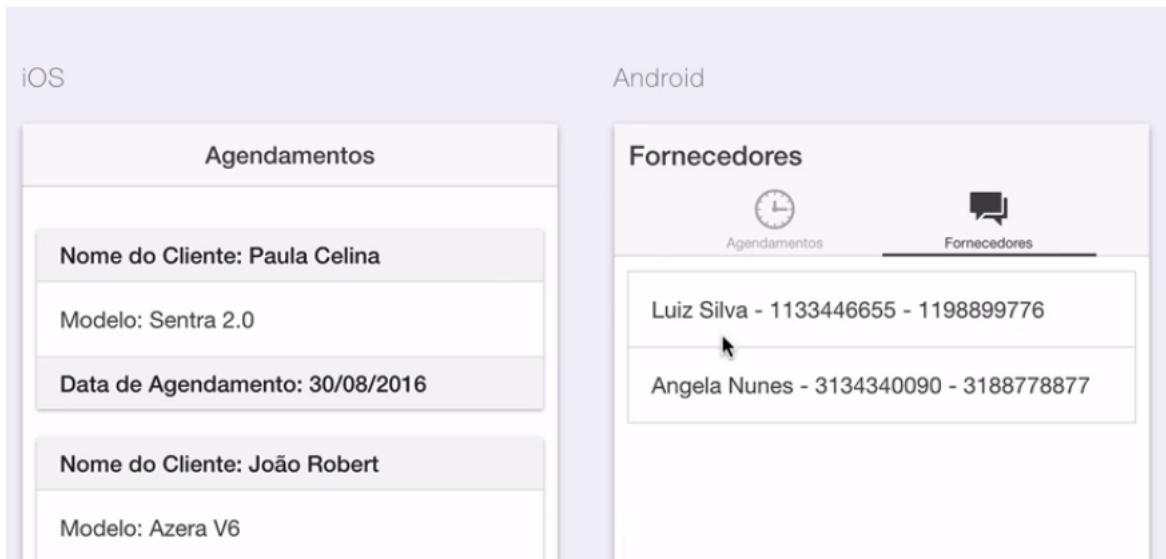
No `fornecedores.html`, queremos que o `item` se repita dependendo da quantidade que tiver no Back-End. Removeremos o `id` criado por ele, porque será desnecessário.

```
<ion-view title="Fornecedores" id="page3">
  <ion-content padding="true" class="has-header">
    <ion-list id="fornecedores-list1">
      <ion-item ng-repeat="fornecedor in fornecedores" ></ion-item>
    </ion-list>
  </ion-content>
</ion-view>
```

Em seguida, adicionaremos os dados aqui. Adicionaremos `nome`, `telefoneComercial` e o `telefoneCelular`, separados por um hífen (-).

```
<ion-view title="Fornecedores" id="page3">
  <ion-content padding="true" class="has-header">
    <ion-list id="fornecedores-list1">
      <ion-item ng-repeat="fornecedor in fornecedores" > {{fornecedor.nome}} - {{fornecedor.telefoneComercial}} - {{fornecedor.telefoneCelular}}
    </ion-list>
  </ion-content>
</ion-view>
```

No browser, veremos como ficou a tela de fornecedores.



Deixarei como exercício para você colocar o telefone com a formatação da `lib` que utilizamos anteriormente e deixar o telefone da seguinte forma:

Luiz Silva - (11)3344-6655 - (11)98889-9776

Nossa aplicação ficou inteligente e bem simples - era este o nosso objetivo.

Usamos o Ionic Creator para gerarmos o protótipo e usamos a inteligência por trás dela para pegarmos os dados do Back-End.

Até a próxima!